1~3周学习报告

这几周主要学习了java语言的基础内容以及对JavaScript的复习

1. 学习了java语法的基本结构以及基本数据类型
2. 学习了java语言如何定义类与如何创建对象

类：类是一个模板，它描述一类对象的行为和状态。

对象：对象是类的一个实例（对象不是找个女朋友），有状态和行为。

一个类可以包含以下类型变量：

局部变量：在方法、构造方法或者语句块中定义的变量被称为局部变量。变量声明和初始化都是在方法中，方法结束后，变量就会自动销毁。

成员变量：成员变量是定义在类中，方法体之外的变量。这种变量在创建对象的时候实例化。成员变量可以被类中方法、构造方法和特定类的语句块访问。

类变量：类变量也声明在类中，方法体之外，但必须声明为 static 类型。

构造方法

每个类都有构造方法。如果没有显式地(明确的写出)为类定义构造方法，Java 编译器将会为该类提供一个默认构造方法(无参的构造方法)。

在创建一个对象的时候，至少要调用一个构造方法。构造方法的名称必须与类同名，一个类可以有多个构造方法。

下面是一个构造方法示例：

public class Puppy{

public Puppy(){

}

public Puppy(String age){

// 这个构造器仅有一个参数：age

}

//一个类可以同时给出多个构造方法

Public Puppy(int age){

}

//构造方法也可以重载

}

创建对象

对象是根据类创建的。在Java中，使用关键字 new 来创建一个新的对象。创建对象需要以下三步：

声明：声明一个对象，包括对象名称和对象类型。

实例化：使用关键字 new 来创建一个对象。

初始化：使用 new 创建对象时，会调用构造方法初始化对象。

下面是一个创建对象的例子：

public class Puppy{

String name;

public Puppy(String name){

//这个构造器仅有一个参数：name

System.out.println("小狗的名字是 : " + name );

}

public static void main(String[] args){

// 下面的语句将创建一个Puppy对象

Puppy myPuppy = new Puppy( "tommy" );

Puppy.name; //访问实例化成员变量

}

}

Java面对对象的三大特征：封装，继承，多态

封装：Java中的封装是指一个类把自己内部的实现细节进行隐藏，只暴露对外的接口（setter和getter方法）。封装又分为属性的封装和方法的封装。把属性定义为私有的，它们通过setter和getter方法来对属性的值进行设定和获取。

继承：Java中的继承是指在一个现有类（父类）的基础上在构建一个新类（子类），子类可以拥有父类的成员变量以及成员方法（但是不一定能访问或调用，例如父类中private私有的成员变量以及方法不能访问和调用）。继承的作用就是能提高代码的复用性。子类拥有父类中的一切（拥有不一定能使用），它可以访问和使用父类中的非私有成员变量，以及重写父类中的非私有成员方法。

多态：Java中的继承是指在一个现有类（父类）的基础上在构建一个新类（子类），子类可以拥有父类的成员变量以及成员方法（但是不一定能访问或调用，例如父类中private私有的成员变量以及方法不能访问和调用）。继承的作用就是能提高代码的复用性。子类拥有父类中的一切（拥有不一定能使用），它可以访问和使用父类中的非私有成员变量，以及重写父类中的非私有成员方法。

1. Java中重要的修饰符：

权限修饰符： private 默认 protected public 四种

同一个类中四种都可以访问

同一个包中子类，无关类可以访问后三种

不同包的子类可以访问后两种

不同包的无关类只能访问 public所修饰的

static关键字：是静态的意思，可以修饰成员方法，成员变量

static修饰的特点：

被类的所有对象共享(这也是我们判断是否使用静态关键字的条件：对象的相同变量值是否相同赋值)，

只要有一个对象进行了赋值，其余的对象的对应值也会改变

可以通过类名调用，也可以通过对象名调用（一般采用类名.变量名调用）

Final修饰符：final关键字是最终的意思，可以修饰成员方法，变量，类

特点：

修饰方法：表明该方法是最终方法，不可以被重写

如父类中有方法 public final void show(){},则子类不可以去重写show方法，不过依然可以调用 super.show()

修饰变量：表明该变量是常量，不能再被赋值

final int age = 10;说明这是一个常量，不能再被改变

修饰类：表明该类是最终类，不能被继承

public final class Fu{} 代表该类是最终类，不能被继承如：public Zi extends Fu{}

final 修饰基本类型：指修饰的基本类型的值不可以改变

final 修饰引用类型：指修饰的引用类型的地址不可以改变，但是地址里的内容(成员变量的值)可以改变

包的概述和使用

包其实就是文件夹 作用：对类进行分类和管理

包的定义格式： package 包名 （多级包用 . 分开） 在代码开头书写

带包的java类编译执行

手动建包：

按照以前的格式编译java文件 javac HelloWorld.java

手动创建包 在对应盘建立文件夹(对应的包名，多级就分多级来创建，每一级之间用 . 隔开)

把class文件放到包的最里面 把helloworld.class文件放到对应文件夹下

带包执行 java com.ithheima.HelloWorld

自动建包： javac -d.HelloWorld.java 然后进行最后带包执行

导包得概述和使用

使用不同包下的类时，使用的时候要写类的全部路径

如：Inherit.Inherit i = new Inherit.Inherit();

而且每次调用不同包类时都要再写一次，太麻烦

为了简化导包的操作，java提供了导包得功能

格式： import包名

举例： import Inherit.MethodRewrite

//导包使用的类只能是public类

抽象类

在java中，一个没有方法体的方法应该定义为抽象方法，而类中如果有抽象方

法，该类必须定义为抽象类。

抽象类的特点：

抽象类和抽象方法必须使用abstract关键字修饰

public abstract class.. public abstract void show(){}

抽象类中不一定有抽象方法，有抽象方法一定是抽象类

抽象类不能实例化

如何实例化：参照多态的方式，通过子类对象实例化，这叫做抽象类多态

抽象类的子类：要么重写抽象类中的所有的抽象方法，要么是抽象类

public abstract void show();

方法没有方法体{} ,会报错，要么有方法体，要么定义为抽象方法

在方法前加 abstract 表示是抽象方法，抽象方法要定义在抽象类里，所以类也

需要加abstract

//抽象类的成员特点

//成员变量：可以是变量，也可以是常量

//构造方法：有构造方法，但是不能实例化

//作用是用于子类访问父类数据的初始化

//成员方法：可以有抽象方法：限定子类必须完成某些动作 也可以有非抽象方

法，提高代码的复用性